Package ‘bravo’

November 12, 2021

Type Package

Title Bayesian Screening and Variable Selection

Version 2.1.2

Author Dongjin Li [aut, cre], Somak Dutta [aut], Vivekananda Roy [ctb]

Maintainer Dongjin Li <liyangxiaobei@gmail.com>

Description Performs Bayesian variable screening and selection for ultra-high dimensional linear regression models.

License GPL-3

Encoding UTF-8

Depends R (>= 3.0.2), methods

Imports Rcpp (>= 1.0.2), Matrix (>= 1.2-17)

LinkingTo Rcpp

NeedsCompilation yes

RoxygenNote 7.1.1

Repository CRAN

Date/Publication 2021-11-12 05:20:02 UTC

R topics documented:

bits — *Bayesian Iterated Screening (ultra-high, high or low dimensional).*

Description

Perform Bayesian iterated screening in Gaussian regression models

Usage

```
bits(X, y, lam = 1, w = 0.5, pp = FALSE, max.var = nrow(X))
```

Arguments

`X`	An $n \times p$ matrix. Sparse matrices are supported and every care is taken not to make copies of this (typically) giant matrix. No need to center or scale.
`y`	The response vector of length `n`.
`lam`	The slab precision parameter. Default: `1`.
`w`	The prior inclusion probability of each variable. Default: `1/2`.
`pp`	Boolean: If `FALSE` (default) the algorithm stops after including `max.var` many variables. If true, the posterior probability stopping rule is used.
`max.var`	The maximum number of variables to be included.

Value

A list with components

`model.pp`	An integer vector of the screened model.
`postprobs`	The sequence of posterior probabilities until the last included variable.
`lam`	The value of lam, the slab precision parameter.
`w`	The value of w, the prior inclusion probability.

References

Wang, R., Dutta, S., Roy, V. (2021) Bayesian iterative screening in ultra-high dimensional settings. https://arxiv.org/abs/2107.10175

Examples

```
n=50; p=100;
TrueBeta <- c(rep(5,3),rep(0,p-3))

rho <- 0.6
x1 <- matrix(rnorm(n*p), n, p)
X <- sqrt(1-rho)*x1 + sqrt(rho)*rnorm(n)
y <- 0.5 + X %*% TrueBeta + rnorm(n)
res<-bits(X,y, pp=TRUE)
res$model.pp # the vector of screened model
res$postprobs # the log (unnormalized) posterior probabilities corresponding to the model.pp.
```

`predict.sven` *Make predictions from a fitted "sven" object.*

Description

This function makes point predictions and computes prediction intervals from a fitted "sven" object.

Usage

```
## S3 method for class 'sven'
predict(
  object,
  newdata,
  model = c("WAM", "MAP"),
  interval = c("none", "MC", "Z"),
  return.draws = FALSE,
  Nsim = 10000,
  level = 0.95,
  alpha = 1 - level,
  ...
)
```

Arguments

`object`	A fitted "sven" object
`newdata`	Matrix of new values for `X` at which predictions are to be made. Must be a matrix; can be sparse as in Matrix package.
`model`	The model to be used to make predictions. Model "MAP" gives the predictions calculated using the MAP model; model "WAM" gives the predictions calculated using the WAM. Default: "WAM".
`interval`	Type of interval calculation. If `interval = "none"`, only point predictions are returned; if `interval = "MC"`, Monte Carlo prediction intervals are returned; if `interval = "Z"`, Z prediction intervals are returned.
`return.draws`	only required if `interval = "MC"`. if `TRUE`, the Monte Carlo samples are returned. Default: `FALSE`.
`Nsim`	only required if `interval = "MC"`. The Monte Carlo sample size. Default: 10000.
`level`	Confidence level of the interval. Default: 0.95.
`alpha`	Type one error rate. Default: 1-`level`.
`...`	Further arguments passed to or from other methods.

Value

The object returned depends on "interval" argument. If `interval = "none"`, the object is an `ncol(newdata)` $\times 1$ vector of the point predictions; otherwise, the object is an `ncol(newdata)` $\times 3$ matrix with the point predictions in the first column and the lower and upper bounds of prediction intervals in the second and third columns, respectively.

if return.draws is `TRUE`, a list with the following components is returned:

`prediction`	vector or matrix as above
`mc.draws`	an `ncol(newdata)` $\times$ `Nsim` matrix of the Monte Carlo samples

Author(s)

Dongjin Li and Somak Dutta
Maintainer: Dongjin Li <dongjl@iastate.edu>

References

Li, D., Dutta, S., Roy, V.(2020) Model Based Screening Embedded Bayesian Variable Selection for Ultra-high Dimensional Settings http://arxiv.org/abs/2006.07561

Examples

```
n = 80; p = 100; nonzero = 5
trueidx <- 1:5
nonzero.value <- c(0.50, 0.75, 1.00, 1.25, 1.50)
TrueBeta = numeric(p)
TrueBeta[trueidx] <- nonzero.value

X <- matrix(rnorm(n*p), n, p)
y <- 0.5 + X %*% TrueBeta + rnorm(n)
res <- sven(X=X, y=y)
newx <- matrix(rnorm(20*p), 20, p)
# predicted values at a new data matrix using MAP model
yhat <- predict(object = res, newdata = newx, model = "MAP", interval = "none")
# 95% Monte Carlo prediction interval using WAM
MC.interval <- predict(object = res, model = "WAM", newdata = newx, interval = "MC", level=0.95)
# 95% Z-prediction interval using MAP model
Z.interval <- predict(object = res, model = "MAP", newdata = newx, interval = "Z", level = 0.95)
```

`sven` *Selection of variables with embedded screening using Bayesian methods (SVEN) in Gaussian linear models (ultra-high, high or low dimensional).*

Description

SVEN is an approach to selecting variables with embedded screening using a Bayesian hierarchical model. It is also a variable selection method in the spirit of the stochastic shotgun search algorithm. However, by embedding a unique model based screening and using fast Cholesky updates, SVEN produces a highly scalable algorithm to explore gigantic model spaces and rapidly identify the regions of high posterior probabilities. It outputs the log (unnormalized) posterior probability of a set of best (highest probability) models. For more details, see Li et al. (2020).

Usage

```
sven(
  X,
  y,
  w = sqrt(nrow(X))/ncol(X),
  lam = nrow(X)/ncol(X)^2,
  Ntemp = 3,
  Tmax = (log(log(ncol(X))) + log(ncol(X))),
  Miter = 50,
  wam.threshold = 0.5,
  log.eps = -16,
  L = 20,
  verbose = TRUE
)
```

Arguments

`X`	The $n \times p$ covariate matrix without intercept. The following classes are supported: `matrix` and `dgCMatrix`. Every care is taken not to make copies of this (typically) giant matrix. No need to center or scale this matrix manually. Scaling is performed implicitly and regression coefficient are returned on the original scale.
`y`	The response vector of length n. No need to center or scale.
`w`	The prior inclusion probability of each variable. Default: $sqrt(n)/p$.
`lam`	The slab precision parameter. Default: n/p^2 as suggested by the theory of Li et al. (2020).
`Ntemp`	The number of temperatures. Default: 3.
`Tmax`	The maximum temperature. Default: $\log\log p + \log p$.
`Miter`	The number of iterations per temperature. Default: `50`.
`wam.threshold`	The threshold probability to select the covariates for WAM. A covariate will be included in WAM if its corresponding marginal inclusion probability is greater than the threshold. Default: 0.5.
`log.eps`	The tolerance to choose the number of top models. See detail. Default: -16.
`L`	The minimum number of neighboring models screened. Default: 20.
`verbose`	If `TRUE`, the function prints the current temperature SVEN is at; the default is TRUE.

Details

SVEN is developed based on a hierarchical Gaussian linear model with priors placed on the regression coefficients as well as on the model space as follows:

$$y|X, \beta_0, \beta, \gamma, \sigma^2, w, \lambda \sim N(\beta_0 1 + X_\gamma \beta_\gamma, \sigma^2 I_n)$$

$$\beta_i|\beta_0, \gamma, \sigma^2, w, \lambda \overset{indep.}{\sim} N(0, \gamma_i \sigma^2/\lambda),\ i = 1, \ldots, p,$$

$$(\beta_0, \sigma^2)|\gamma, w, p \sim p(\beta_0, \sigma^2) \propto 1/\sigma^2$$

$$\gamma_i|w, \lambda \overset{iid}{\sim} Bernoulli(w)$$

where X_γ is the $n \times |\gamma|$ submatrix of X consisting of those columns of X for which $\gamma_i = 1$ and similarly, β_γ is the $|\gamma|$ subvector of β corresponding to γ. Degenerate spike priors on inactive variables and Gaussian slab priors on active covariates makes the posterior probability (up to a normalizing constant) of a model $P(\gamma|y)$ available in explicit form (Li et al., 2020).

The variable selection starts from an empty model and updates the model according to the posterior probability of its neighboring models for some pre-specified number of iterations. In each iteration, the models with small probabilities are screened out in order to quickly identify the regions of high posterior probabilities. A temperature schedule is used to facilitate exploration of models separated by valleys in the posterior probability function, thus mitigate posterior multimodality associated with variable selection models. The default maximum temperature is guided by the asymptotic posterior model selection consistency results in Li et al. (2020).

SVEN provides the maximum a posteriori (MAP) model as well as the weighted average model (WAM). WAM is obtained in the following way: (1) keep the best (highest probability) K distinct models $\gamma^{(1)}, \ldots, \gamma^{(K)}$ with

$$\log P\left(\gamma^{(1)}|y\right) \geq \cdots \geq \log P\left(\gamma^{(K)}|y\right)$$

where K is chosen so that $\log\left\{P\left(\gamma^{(K)}|y\right)/P\left(\gamma^{(1)}|y\right)\right\} >$ `log.eps`; (2) assign the weights

$$w_i = P(\gamma^{(i)}|y)/\sum_{k=1}^{K} P(\gamma^{(k)}|y)$$

to the model $\gamma^{(i)}$; (3) define the approximate marginal inclusion probabilities for the jth variable as

$$\hat{\pi}_j = \sum_{k=1}^{K} w_k I(\gamma_j^{(k)} = 1).$$

Then, the WAM is defined as the model containing variables j with $\hat{\pi}_j >$ `wam.threshold`. SVEN also provides all the top K models which are stored in an $p \times K$ sparse matrix, along with their corresponding log (unnormalized) posterior probabilities.

Value

A list with components

`model.map` A vector of indices corresponding to the selected variables in the MAP model.

model.wam	A vector of indices corresponding to the selected variables in the WAM.
model.top	A sparse matrix storing the top models.
beta.map	The ridge estimator of regression coefficients in the MAP model.
beta.wam	The ridge estimator of regression coefficients in the WAM.
mip.map	The marginal inclusion probabilities of the variables in the MAP model.
mip.wam	The marginal inclusion probabilities of the variables in the WAM.
pprob.map	The log (unnormalized) posterior probability corresponding to the MAP model.
pprob.top	A vector of the log (unnormalized) posterior probabilities corresponding to the top models.
stats	Additional statistics.

Author(s)

Dongjin Li and Somak Dutta
Maintainer: Dongjin Li <dongjl@iastate.edu>

References

Li, D., Dutta, S., Roy, V.(2020) Model Based Screening Embedded Bayesian Variable Selection for Ultra-high Dimensional Settings http://arxiv.org/abs/2006.07561

Examples

```
n=50; p=100; nonzero = 3
trueidx <- 1:3
nonzero.value <- 5
TrueBeta <- numeric(p)
TrueBeta[trueidx] <- nonzero.value

rho <- 0.6
x1 <- matrix(rnorm(n*p), n, p)
X <- sqrt(1-rho)*x1 + sqrt(rho)*rnorm(n)
y <- 0.5 + X %*% TrueBeta + rnorm(n)
res <- sven(X=X, y=y)
res$model.map # the MAP model
res$model.wam # the WAM
res$mip.map # the marginal inclusion probabilities of the variables in the MAP model
res$mip.wam # the marginal inclusion probabilities of the variables in the WAM
res$pprob.top # the log (unnormalized) posterior probabilities corresponding to the top models.

res$beta.map # the ridge estimator of regression coefficients in the MAP model
res$beta.wam # the ridge estimator of regression coefficients in the WAM
```

Index